GREQAM

Groupement de Recherche en Economie
Quantitative d'Aix-Marseille - UMR-CNRS 6579
Ecole des Hautes Etudes en Sciences Sociales
Universités d'Aix-Marseille II et III

Document de Travail
n°2009-45

Firms' contribution to open source software and the dominant user skill.

Nicolas JULLIEN
et Jean-Benoît ZIMMERMANN

March 2009

DT-GREQAM

Firms' contribution to open source software and the dominant user skill.

Nicolas Jullien – Institut Telecom Bretagne & UEB / LUSSI, M@RSOUIN
Jean-Benoît Zimmermann – CNRS / GREQAM and IDEP

This version : March 2009 *

Abstract : *Free, libre or open source software (FLOSS) is nowadays produced not only by individual benevolent developers but, in a growing proportion, by firms that hire programmers for their own objectives of development in open source or for contributing to open source projects in the context of dedicated communities. A recent literature has focused on the question of the business models explaining how and why firms may draw benefits from such involvement and their connected activities. They can be considered as the building blocks of a new modus operandi of an industry, built on an alternative approach to intellectual property management. Its prospects will depend on both the firms' willingness to rally and its ability to compete with the traditional "proprietary" approach. As a matter of fact firms' involvement in FLOSS, while growing, remains very contrasted, depending on the nature of the products and the characteristics of the markets. The paper asks why do for-profit firms contribute to FLOSS development and why some firms contribute more than the others. The common explanation is that FLOSS is often a complement to proprietary software (or hardware or services) that the for-profit firm sells at a positive price. We present an alternative explanation based the users' skill level. When users are skilled, opening the software is likely to result in a better product because the user base will contribute improvements (find bugs, write fixes and produce new features). We introduce the concept of the dominant user's skill and we set up a theoretical model to better understand how it may condition the nature and outcome of the competition between a FLOSS firm and a proprietary firm. We discuss these results in the light of empirical stylized facts drawn from the recent trends in the software industry.*

1. Introduction

"Free"/"libre" or "open source" software (FLOSS) is software whose source-code, that is the explicit expression of the programming work, remains openly accessible. Until recently, FLOSS was considered only to be of interest to programmers motivated by the building and sharing of a base of programs developed for their own needs (Lakhani & von Hippel (2003), Demazière & al. (2006)).

Today FLOSS has apparently become an economic issue of considerable importance. This is remarkably illustrated, in the case of Europe is by the publication of a, important report on this subject by the European Commission at the end of year 2006 [1]. In industrial terms, the open source model has widely stuck

*The authors thank warmly the two anonymous referres for their excellent reports and suggestions and the EMR editors for the very accurate advice they gave us for the improvement of the paper. We are indebted for helpful comments to the participants of the Conferences and seminars where we had the opportunity to present the project of this paper, more particularly the DIME London Conference "Intellectual Property Rights for Business and Society", Birbeck University, 14th and 15th September 2006, the international conference « The diffusion of FLOSS and the Organisation of the Software Industry : From Social Networks to Economic and Legal Models », Nice Sophia-Antipolis, May 31st, June 1st 2007, the Complex Markets Workshop, Warwick Business School, March 31st - April 2d 2008. This research has benefited from the support of the Institut Telecom (France), the French ANR and the European STREP "Complex Markets".

[1] " « FLOSS is good for the European economy, employment and firms competitiveness... » http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf.

out the limits of the computer industry to to affect other IT fields like the telecommunications industry[2]. It appear nowadays to be at stake of very important competition issues as a growing number of industrial actors, including Microsoft itself, have eventually acknowledge. Today, open source software is increasingly integrated into many commercial offers (Novell, buying Ximian and SuSE, Sun open-sourcing its operating system, IBM open-sourcing its development tool software Eclipse, even Microsoft, deciding to distribute some of its software products under open license[3]). This situation, in which commercial business relies on the existence and durability of non-market activities, challenges traditional industrial economic theory.

As in any cooperative agreement devoted to technology or knowledge development, agents pool assets together in a "pre-competitive" phase and share the products of their efforts before returning to competition (Crémer & al. (1990), Bhattacharya and Guriev (2006)). But such agreements remain closed to third parties. On the contrary a FLOSS project is an open contribution game in which the list of players is not bounded ex-ante by a cooperative agreement and the output of which is a public good that cannot be appropriated by any of the players on an exclusive way. This would rather correspond to the formation of a consortium for the production of a standard[4].

Scholars see this phenomenon as an extreme case of 'open innovation'[5] (Chesbrough, 2003), and thus of a laboratory of innovation production in some Internet based/knowledge based industry. This radically brand new way of producing software challenges the foundations and competition regime of the software industry which was evolving more towards strengthening the intellectual property protection (Bessen and Hunt, 2007).Two radically opposed ways of working are confronting each other, based on two opposing views about intellectual property management. On the one hand, the "traditional" position, referred here as the "proprietary" approach, defends the need to strengthen the intellectual property regime to respond to a growing ease with which of software codes can be appropriated and used without any fee[6] . FLOSS defenders, on the other hand, argue that this position is inefficient for innovation dynamics, reinforcing the market power of the sole dominant firms and thus the oligopolistic nature of the industry which is economically inefficient[7].

The result of this confrontation is then of huge importance for the future of the software industry and, more widely, for the whole information and communication industry. From an even broader perspective, this case of open source software can be viewed as a precursor of what could also take place in a wider array of industries where innovation has a strong role to play in competition and requires to access to a larger and larger range of knowledge and resources.

This confrontation appeals for a better understanding of how FLOSS open innovation works, ie how firms turn FLOSS products into business, what is the level of investments firms have to make to exploit this production on the market, and especially how far they have to involve into the cooperative production process to do so.

The paper asks why do for-profit firms contribute to FLOSS development and why some firms contribute more than the others. The common explanation is that FLOSS is often a complement to proprietary software (or hardware or services) that the for-profit firm sells at a positive price. We present an alternative explanation based the users' skill level. When users are skilled, opening the software is likely to result in a

[2] Regarding voice on IP, FLOSS Asterisk, http://www.asterisk.org/, is one of the most popular choices. Motorola has selected Linux as one of three operating systems for its mobile terminals.

[3] http://solutions.journaldunet.com/0404/040407_microsoft.shtml

[4] What we mean is that a player offers a standard by developing a software that the other players can adopt and help to develop. This "unilateral" adoption is usually called 'bandwagon' in the literature on standards (see, for instance, Farrell and Saloner, 1985).

[5] Defined as "a paradigm that assumes that firms can and should use external ideas as well as internal ideas, and internal and external paths to market, as they look to advance their technology" (Chesbrough, 2006).

[6] See, for instance, the arguments developed by the Business Software Alliance (BSA), which regroups the main actors of the software sector, such as IBM, Microsoft, Apple, Sybase, etc. on http://www.bsa.org/, especially their latest white paper (August 2008, http://www.bsa.org/country/Research%20and%20Statistics/~/media/96FC7EAFF3E84436AF62C3B393F207B1.ashx)

[7] See the arguments of the Free Software Foundation (http://www.fsf.org), or Boldrin and Levine (2007) .

better product because the user base will contribute improvements (find bugs, write fixes and produce new features).[8]

In the last decade, an abundant and growing literature has discussed this question. According to us, it can be split into two approaches, two points of views, ie two data sources, the first looking at the production side and the second at the market side to explain firms' interest regarding FLOSS.

The production side starts from the communities and evaluates the level and way of the involvement of firms into them. The research agenda, as put forward by Von Hippel & Von Krogh (2003) is to understand how firms invest themselves into a specific community and what these communities can provide to them. From an organization science point of view, the question is how agents organize themselves to manage distributed innovation and in which conditions firms can capture a part of this innovation for business purpose. The asset of such studies is the availability of data from communities. Their exploitation leads to crucial results for understanding the links between business and open source.

In 2005, Lakhani & Wolf (2005), analysing a survey of 287 communities (ie people active in FLOSS development projects) show the importance of the business participation to FLOSS communities, as "a majority of [their] respondents are skilled and experienced professionals working in IT-related jobs, with approximately 40 percent being paid to participate in the F/OSS project."

Henkel (2006), while studying the "embedded" Linux system, has shown that business involvement pursued several strategies and that firms did not reveal all the codes they produced but rather carefully selected their contributions. Dahlander & Wallin (2006) looking at the "GNOME" graphic interface project, and standing on Teece (1986)'s theory, defend the idea that, hiring developers who participate to this development project, these firms try to control a complementary asset important for building their products and services. This enlightens IBM's strategy of supporting and investing in the development of Linux while selling hardware (mainframe), software (Lotus suite) and services. "A complementary asset that exists outside firm boundaries and outside their ownership or hierarchical control"(Dahlander and Wallin, 2006). This is even the sole control means when innovations, as observed by Von Hippel (2005) are produced by a community of innovative users that protect their innovation from a private appropriation through a GPL license[9].

Iansiti & Richards (2006) identify, amongst the various FLOSS projects a "money-driven cluster" where « IT vendors' motives are economic. In this cluster, significant investments have been made in projects that will serve as complementary assets to drive revenues to vendors' core businesses ».

However, these authors have looked at already well established communities, where the software developed is shared by numerous actors, people or firms. This may hardly explain why some companies, like MySQL AB, which owns the entire eponymous database software , open source it and, still, remain responsible for the majority of its development, as if it was the core asset of their business[10].

On the other hand, from a strategy and management sciences point of view, some scholars have looked at the business side and explained the use of FLOSS products by the characteristics of the market and firms' positioning on this market. The two main questions here are the definition of a business model (what do you sell when choosing FLOSS?), and the links between a business model and the involvement into communities.

Chang & al. (2007) have surveyed the literature on the different "FLOSS business models", and classified them "into five types: (a) Support Contracts; (b) Split Licensing; (c) Community; (d) Valued-added closed source; (e) Macro R&D Infrastructure". They have looked at the advantage and disadvantage of each models, proposing a case study for each, but without investigating the involvement into the communities, and the market condition into which each model is the most efficient.

[8]This concise formulation of our concern has been suggested by one of our anonymous reviewer.

[9] The main principle of the GPL (General Public Licence) is to make its adopters disclose the source-code of the programs concerned and of any further improvement if they circulate them, as well as the free circulation of the code under the sole condition of maintaining its "open" character.

[10] This means that any developer/contributor wanting to make a contribution to the official MySQL product, she has to transfer her copyright to MySQL. http://forge.mysql.com/contribute/cla.php. Once owning the whole copyright, the firm can manage a dual licensing scheme, distributing the product under the license she wants, either GPL or more classical closed license. So a customer that whishes not to reveal further enhancements of the source code has to maintain by herself these enhancements.

Surveying Italian firms, Bonaccorsi, Giannangeli & Rossi (2006), have proposed a definition of "FLOSS based business" and different reasons for firms participating to FLOSS development. But they have not really explained the link between the kind of business model and the degree of involvement into communities.

The first to have set up this link between these two aspects are Finnish scholars. Surveying Finnish firms, Dahlander and Magnusson (2005, 2008) have shown that, in a large extent the variety of firms' involvement into FLOSS can be understood in the light of the place of the software in the firm's business model (as core activity or not). Still on Finish firms data, but at product level, Kosky (2007) has remarked that "it seems that factors other than those typically found to explain differences in entrepreneurial innovation behaviour such as firm size and age account for the differences in the product and license type strategies of the software companies. [His] data indicate that the firm ownership structure has a major influence for the software firms' product-level business strategies" (p. 123).

If these studies prove the link between the market and the involvement into FLOSS, they suffer from some limitations. They are rather descriptive and do not propose explanations of the variation of firms' behaviour in similar markets: why Asus or Dell put Linux on their computer without participating to the development when HP or IBM do? This may be because these authors look at firm's level without trying to look at the differences between the different branch of the industry, as West (2003) could have done on the server/operating system market (what he called "platform market").

So, our present work belongs to this second category of approaches It tries to systematize West's approach to the whole Information Technology industry . Our major concern is the following: how could we explain that, in similar branches regarding the place of the software, or the network effects (like business solution markets), there are sometimes a bunch of market successful firms sponsoring FLOSS offers (such as content management systems, or data base tools) and in others no market successful one (like for business intelligence).

The argument we will put forward in this paper is that these differences can be explained by the characteristics of the demand, and more particularly of the users skills. As Von Hippel (1988) explained, when clients are skilled users, capable to express their needs and even to develop prototypes, they are in the best position to propose innovations, even if firms are needed to industrialize them. To this argument we will add that when users, in the opposite situation, are unskilled and do not understand source code, they rather tend to assimilate the advantages of a free software with those of a freeware, that is obtainable free of charge.

The rest of the paper is organized as follows. In section 2, using stylised facts, we show how the level of skills of the users can conditions the nature and issue of the competition in the computer industry and, thus the place of FLOSS strategies, bringing us to introduce the concept of "dominant user's skill". In section 3, to formally show the implications of this demand characteristic on the firm's choices, we present a formal competition model in which two firms, one open-source and one proprietary compete for a market characterized by a given distribution of users' skills. In a conclusive section 4 we discuss the lessons and the managerial and industrial economics implications of the model outcomes and we conclude on the limits and further extensions of the paper..

2. IT industry, attitudes toward FLOSS, and the dominant user's skill.

1. FLOSS involvement and the role of the users.

Since the mid-1990s, a growing number of commercial firm, either new entrants or incumbents, have decided to integrate FLOSS products in their own specific offer or toolboxes, even investing by different means in FLOSS development. Of course these new emerging strategies must be understood in the light of IP protection prevailing in each market segment and the need to strengthen competitive advantages or to rely on new ones.

Regarding the degree of involvement in FLOSS dynamics, the more active actors seems to be found in sectors where software development and use is either a core activity or a crucial condition for performances, as it is the case for server manufacturers or architects of information systems (adoption of Linux by IBM, HP since the beginning of the years 2000). At the other extreme, the weakest involvement is found amongst hardware suppliers that can only feel concerned by FLOSS for compatibility and price purposes.

When FLOSS adoption is related to marginal aspects of differentiation, it seems to have little impact on industrial structure and competition. This is generally the case for most of hardware producers, when hard-soft-content is no longer bundled (servers, computers, Personal Communication Tools, DVD and MP3 players...)

Surprisingly, FLOSS diffusion impacts mainly firms in software based industries. This has to be understood regarding how their core competences have evolved and shifted significantly. Their main challenge is less and less to supply a "software solution" to a given problem at a given time, but increasingly to deal with short to long term uncertainty over IT system production and management. Users ask for solutions able to protect them against uncertainty, granting interoperability, bug resolution, the satisfaction of new needs and the integration of technical advances . The trade-off between available solutions is not posed in terms of their cost of acquisition but of their "TCO" (total cost of ownership), in which the future costs and the costs for granting interoperability and adaptability have to be estimated. This is precisely what architects, business programs and platform producers sell to skilled users, aware of these problems and signals. On these markets the FLOSS organization seems to represent an asset for producers, who can display their involvement and succeed in building sustainable business models (see the examples of RedHat, MySQL or, in France Linagora). But, as explained before, this is only an asset if the market regards FLOSS as providing a value added to the product, i.e. if this brings the users a potential for increasing their utility.

How and why may those different users contribute directly or indirectly to FLOSS projects? First of all, contribution does not necessary imply code development but can take various forms in the product development and improvement. Users have to be considered as valuable "sources of innovation" (Von Hippel), not only for program testing and debugging but also for improving the product usability and performances. People decide to contribute if they get interested by the product, or if they have a problem, in which case they can either report the problem directly or through an intermediary, the supplier for instance, that allows the user to pass from a passive to an active use of the project.

Actually, the users, understood as the persons choosing the solution (thus not always being the "end-users"), are rather different from one market to another, causing the competitive advantage to rely on different features.

Let us distinct three main types of users according to their relation to the product and the technology (Zimmermann 1995, Kogut and Metiu 2001, von Hippel 1988, 2002). The first is the category of "Naïve customers or users" (that we denote N) who are not endowed with noticeable technical skills and do not individually weigh very much in economic terms. The second is the category of "Kogut-Metiu Users" (KM)[11] who are not able to contribute to software development but can generate new features or innovations by revealing their own needs. Above all, they represent an irreplaceable testing and debugging base. KM users are sensitive to price and quality arguments The third category is that of the "Von Hippel Users" (VH) who act as "sources of innovation" (Von Hippel, 1988) able to contribute to software development by proposing improvements or modifications, developing it by themselves or at least able to design the technical specifications.

Users play a double role, deriving from both their economic and technical standing. Depending on the market, and especially their bargaining power in it, the users are more or less able to select the (technical) offers. At one extreme, users and contracts in the global service/architects market are related to large structures, with substantial buying capacities and generally endowed with significant technical skills. So they are likely to influence economic and technical choices. At the other extreme low price computers address a mass market where individual users, in their vast majority have little budget and/or few skills. Their influence on market evolution is negligible at an individual level but of global importance in terms of elasticity to prices. But this analysis should be nuanced in the case of intermediation by a "prescriber", who

[11] In reference to the notion of « frontier-users » put forward by Kogut and Metiu (2001)

orders and defines the characteristics for a large number of machines, destined for mass distribution by his own means (local government for secondary schools in France[12], education in rural area in developing countries[13], ...) That's the reason why, when speaking about the "user", we mean the person who negotiates or chooses the characteristics of the good, who is not always the end user.

Of course different types of users are co-existing in any given market. But the dispersion of users' skills in the related technology and more particularly in software doesn't follow the same distribution from one segment to another. Even if skilled users are likely to be found in any market, they may represent a too small share to play a significant role in it and catch the interest the concerned firms for their specific demand. Conversely, thanks to the Internet, a handful of very talented users around the world can weigh enough together to develop a FLOSS alternative to private offers and contribute to the emergence of a FLOSS business offer. So, what we denote users' skills appears as a subtle mix between competences and number, from wich could yield a weighted sum of competences.

2. What we can learn from the markets?

There is a wide diversity of actors in the industry in terms of both products and size. Successive waves of innovations and company strategies have led to a progressive reshaping of the industry borders and structure. For example, Internet has impacted the software production, pushing firms to integrate more services in their offers, designing new ways of selling software bases applications, such as Saas (software as a service) (Cusumano, 2004, pp 86-127; Campbell-Kelly & Garcia-Swartz, 2007), However, the foundations of the industry have remained unchanged, since those described by Gérard-Varet & Zimmermann (1985), Zimmermann (1995), Steinmueller (1996), and Cusumano (2004): IT products are built by assembling hardware and software units in a given architecture, and these products (isolated or integrated into networks) are used as parts of information systems and solutions. On the basis of such technical organization, it is then possible to distinguish three large types of "vertical specialization": i. component producers, ii. computers and IT devices suppliers, iii. software editors and service companies providing applications.

All these segments are concerned with software production, as even chipset manufacturers have to deal with the operating systems embeded in the machine integrating their component. They provide drivers for these operating systems, and their incentive to use and develop FLOSS drivers for free operating systems (such as Linux) is a growing function of such systems market size. Since the beginning of the 2000s, some firms like ATI indeed offer such compatible drivers. But, this remains a marginal contribution, and should not have any immediate serious impact on the structure of the FLOSS development organization. So we will not investigate further the strategies towards FLOSS in this segment of the industry.

Remain what is traditionally defined as the hardware part (the machines) and the software part (software and services), with, in between the operating system.

The hardware.

Hardware is increasingly various, from mainframes to netbooks, and from dedicated devices (personal communication tools, video game or music players) to the "swiss knife machines" which are modern computers.

Looking at these markets from the dominant user skill prism helps to understand the adoption of FLOSS within the industry.

1. In the **servers market**, producers have habitually provided proprietary solutions with proprietary Unix[14]. Here suppliers are dealing with highly-skilled VH clients that can make an essential contribution in the context of FLOSS opening. The rise of PC servers has permitted some users to avoid such a bundling problem; moreover, using Linux allows a cheaper offer (vertical advantage) reusing Unix programs

[12] With the aim to provide "a computer for each pupil": http://www.ordina13.com/, http://www.ordi35.fr/

[13] See, for instance, the competition between Microsoft and Mandriva to supply 17,000 computers in Nigeria. http://www.computerworlduk.com/management/government-law/public-sector/news/index.cfm?newsid=6124

[14] See West (2003) for a full discussion of FLOSS strategies in that sector .

(content) portfolio. Thus some firms have been able to widen the servers market from VH users capable of managing their systems by themselves to KM clients, sensitive to prices, but also to the quality of a PC server fitted out with Linux. So new entries have been experienced like the Cobalt[15] one, but the main actors of the Unix "world" have also rapidly developed their own offers, cutting down the sources of vertical differentiation[16].

2. The segment of netbooks, and **low price computers** (LPC) is a mass market where naïve clients are the driving force behind demand, and competition is overall based on prices. When Asus entered the market with its eee-PC, it used Linux for price reasons, because Microsoft Windows Vista was too costly in terms of resources needed and price to be competitive. Since, considering the success of this market, Microsoft has designed a specific, downgraded version of Windows XP for these computers[17]. It is worth noting that, since the middle of 2007, Dell proposes Ubuntu Linux distribution on ones of its first price laptops[18].

3. Between these two cases there is the **high quality computers** (HQC) market, ie computers for firms or computers used to play games, computers requiring good, up-to-date performances. In that segment, exigent users, or frontier, KM users seems to be dominant. It is worth noting that in this desktop market, the main push in favour of open source, for the time being, is driven by organizations or institutions (which we consider as VH users) that take decisions to equip a large number of end-users. Examples are the French "Assemblée Nationale" (French Congress) that has contracted with a service company to install Linux on all the computers provided to MPs[19], or the initiatives of the Nigerian[20] and Macedonian[21] governments for schools, or in the industry, the French automaker Peugeot[22].

So, today, HQC producers may find it hard to switch from Windows to Linux, because this would mean either acquiring new skills (OS management and improvement), or sub-contracting this maintenance to Linux editors (RedHat, SuSE,...) which may lead to another dependence and to conflict relations with the dominant provider. Nevertheless, a possible future evolution in this sense is likely to arise from the pressure of corporate and VH customers becoming more aware of the potentialities of switching to FLOSS. It is worth noting that the Linux offered by HP is part of the enterprise offers branch[23]. In the near future most of the HQCs will probably switch to debundling their machines from the associated OS, to segment more their offer between VH users with the Linux offer and KM users with Windows.

4. **Dedicate digital devices** represent another intermediate case with less skilled customers (KM+N) and a weak degree of involvement on the part of commercial actors into FLOSS, and mainly for compatibility and absorptive capacity purpose.

At one extreme, in the games consoles segment but also to a lesser extent in the music player market, proprietary formats have introduced, a strong bundle of hardware-software-content and FLOSS products are non-existent. Thanks to the MP3 standard or new existing or emerging open standards like Ogg, new entries are always possible in segments like the music players market, but the main actors, like Apple, remain on a strict proprietary strategy. On the contrary, barriers remain high on the video game players market due to the scarcity of independent games capable of running on Linux, unlike the PS2, Xbox and other proprietary standards games. Moreover, when they exist, such games seem harder to obtain for simple users.

---

[15] Cobalt was bought by SUN, which dissolved the products into its own offer. See http://www.sun.com/hardware/serverappliances/eol.html

[16] It worth noting that, on the contrary, SUN, being the leader on the UNIX market, has been reluctant to adopt Linux and is today the server constructor which has the most difficulties to adapt its business model, with recurrent losses.

[17] Eee-PC has been the "most wanted 2007 Christmas gift", according to the constructor, http://eeepc.asus.com/global/

[18] http://www.dell.com/content/topics/segtopic.aspx/linux_3x?c=us&cs=19&l=en&s=dhs

[19] http://www.zdnetasia.com/news/software/0,39044164,61970345,00.htm

[20] http://www.zdnet.co.uk/talkback/0,1000001161,39290511-39001070c-20088736o,00.htm .

[21] http://www.linuxtoday.com/infrastructure/2007091902626NWDPPB

[22] http://www.informationweek.com/news/management/showArticle.jhtml?articleID=201400082

[23] http://h71028.www7.hp.com/enterprise/cache/309906-0-0-0-121.html

On the contrary, there are lots of FLOSS products for **Personal Communication Tools**, or Mobile Computers[24]. Some are proposed by VH users, other by the constructors: :

- if the leader, Nokia only soldl an Internet tablet based on Linux and a development community[25], there are lots of open-source projects around Symbian (partly owned by Nokia, partly by Sony-Ericsson)[26], mainly dedicated to tools for developing applications (libraries, development tools, etc.) and Samsung proposes the first smart phones based on Linux[27];

- the PDA Operating system editor Palmsource is working on the integration of its product on a Linux kernel on its products[28].

For the same reasons as for PC computers, we hardly see naïve or KM people switch from an installed operating system to a FLOSS one. So constructors will continue to drive the market and decide what they integrate in their offer. Implementing Linux on PCT devices may appear as a good strategy to limit differentiation to the core competences of the manufacturers. Operating systems are not at the heart of the products differentiation which is more based on ergonomic aspects and hardware characteristics. In the absence of a still established *de facto* standard, as it stands in the PC market, Linux is to be considered by PCT suppliers, as it is free of charge and benefits from a community of developer-users capable to develop new features and new products outside any proprietary control. In fact, similarly to the PC market, the challenge is the choice of a platform (Operating System) to build the product. Palm is also good example of a company which after having sold its OS division, is now turning toward Linux.

The software.

According to Cusumano (2004, chap. 2), this market can be splited between service and product, and for the product side between business specialized offers and global, platform offers. We will follow this distinction here.

1. In the **software platform market**, the Linux distribution market is another very good illustration of the key role of the demand. Linux publlishers, like RedHat, SuSE, Mandriva (formally Mandrakesoft), have been among the first commercial actors to enter the market using FLOSS. This could appear to be obvious on a mass market with rather naive users and a significant price-based competition. But today, the retail store sales of OS packages represent a negligible part of the revenue of such firms[29], and a major part is targeted to the business market.

One might explain this fact by the development of broadband connection thanks to ADSL. But we believe a more important explanation lies in the skills of the users and the construction of the offer. Consumers buy computers with an OS already installed and few of them are skilled enough to install a different one. Additionally there are no incentives to do so because the pre-installed OS has already been paid for with the computer. So, the diffusion of FLOSS OS on desktop/laptop PCs depends more on the strategies of constructors, as discussed above, than on direct installation by users. And for VH people wanting to install Linux on their PC, other, more technically oriented distributions exist, like Debian, and there is no need to pay for these distributions, available for download on the Web.

On the emerging OS for PC server market, things work differently. Most of the users, of VH or KM type, are aware of the technical questions involved in installing and configuring an OS. It is also easier to buy a

[24] See, for instance, http://tuxmobil.org/ a web site dedicated to Linux and mobile computers.

[25] Nokia 770 Internet Tablet: http://www.nokiausa.com/770/1,7841,feat:1,00.html. Development community: http://www.maemo.org/

[26] In June 2008, Nokia announced to be acquiring the whole share of Symbian and open source it under Eclipse license. See the Symbian foundation Web site: http://www.symbianfoundation.org/

[27] http://linuxdevices.com/news/NS2854558742.html

[28]Palm and Linux: http://news.com.com/2102-1041_3-6175171.html?tag=st.util.print. The web site dedicated by Palm to open source: http://www.palmsource.com/opensource/.

[29] RedHat stopped this activity (see financial report 2006, p. 31); the consumer market (including distributors, OEM sales, e-commerce and Club) represented 2.54M€ (45% of the total earnings) showing a 23.4% decrease for Mandriva 2005-2006 fiscal year; SuSE has been bought by Novell, so these revenues are diluted.

machine without an operating system installed, and the relative price of the OS is lower. FLOSS gives them access to a cheaper but also more open and more adaptable Unix-like operating system, than they could find in the traditional Unix offer. This gave FLOSS OS publishers an undeniable competitive advantage, at least until that server constructors started to offer PC servers with Linux.

2. In the **business software market**, the more skilled the users are, in terms of software development skills, and (although this is a lesser driving force) in terms of cexpressing functionality requirements, the more FLOSS concepts and industrial related offers are likely to spread.

It is clear that the use of open source business software, enabling savings on the cost of licenses, offers a price advantage. Moreover, the fact that the customer can evaluate the product without buying a license is also an advantage in terms of dissemination. It may even be compulsory when dominant players already exist on the market (such as the database market where MySQL proposes software products competing against those of Oracle, IBM and Microsoft who represent more than 80 % of the market) or when customers are highly sensitive to price (such as the ERP market which increasingly concerns SMEs and where open-source products like ERP5 or tiny ERP are now available). This strategy also enables the association of a corporate brand with a product, therefore increasing the notoriety of the firm through distribution of the latter. Moreover, on these technical markets, especially when the customers are developers, availability of the code promotes cooperation. The producer approves the contributions, ensures stability of the tool and helps developers to use it. If some individual contributor becomes important (in terms of contribution volume/quality/innovative aspect), s/he may be hired by a producer, with reduced recruitment costs and risks (ACT or MySQL but also some small services companies are using this method). By contributing to innovation, the developers (and possibly companies using the tool), are therefore guaranteed that their needs will be taken into account more quickly and integrated into the product (which is a fundamental factor in reducing costs, according to Von Hippel 1988).

Obviously, capitalizing on existing products is more difficult, even if, as Muselli (2002) explained, with the entire control of the software , a dual license strategy can be set up to sell the program when requested by the customers (because, for example, they want to integrate it in a larger, closed, package). This is what companies like Qt or MySQL offer. But, today, the main source of revenue again comes from services, more precisely what we call the “3A services” (assistance, assurance and adaptation to the use). Otherwise, adaptation services must be significant enough to finance development of the product. Therefore, the objective is to transform a handicap (significant investments) into a commercial advantage, by increasing the business feedback from users and by considering openness as a way to reduce transaction costs and as a signal of quality. Currently, the main evolution for those firms is to switch from a demand pull strategy (functionalities are developed to stimulate/create the demand) to an 'on-demand' development (development when required and paid for or carried out by the users).

This explains why open source business products are developed mainly in “business” software (ERP, computer infrastructure software like compilers), where users ready to pay for configuration, maintenance or assistance services are numerous. But the scope could easily extend to many technical/professional software activities.

3. As far as the services of the “**architects” market** is concerned, as Horn (2004) points out, assembling components requires access to the source codes (problem of compatibility), and their adaptation to different needs (of users and other components). They must be available in the form of open-source software (therefore legally modifiable).

The competitive advantage in using free software, in addition to price, is therefore the ability to offer an assembled set of components with greater interoperability, which should increase the quality of the final product, on a market where the quality of services is one of the recurrent problems (see De Bandt, 1995). Revenues are generated by assembling and adaptation services, as is the case for any traditional service company.

The only uncertainty about the model concerns the availability of the components: who will develop them and who will maintain them? Moreover, the customers of these companies may already have (proprietary) programs installed that need to be taken into account. In the end, an open source strategy could even be a

guarantee of means (maximum use of free software), but not a guarantee of the results (use of only free software), unless the customer requests this, since in this situation, he keeps the last word.

Two kinds of firms use FLOSS today : newcomers who specialize in FLOSS architecture, using FLOSS as vertical (price) and horizontal differentiation asset, and incumbents, such as IBM for its service activities[30]. Traditional service firms like Cap Gemini are more agnostic with regard to the technologies used and the intellectual property regime involved. They will generally follow the customers' demand which depends on their ability to keep up with the development of the project. These customers are most often large organizations, skilled computer users that are receptive to the opportunity to integrate the most advanced software components, developed under open licenses. So they are becoming increasingly involved in FLOSS as the market grows and matures[31].

Table 1 below summarize the main type of users likely to be found in each sub-sector of the IT industry.

Table 1. The dominant user type in each IT sub-sector.

Actors/ products	***Dominant user type***	***Comments***
Components	VH	Component producers supply hardware manufacturers, aware of the quality and quality-price aspects of the components they will use, as well as the effects of brand reputation of these latter as a signal of quality for their own products.
Servers	VH	The clients are computer-literate people, able to express needs in technical terms, to develop software for their own needs, and to innovate by themselves.
High Quality Computers	KM	HQC users are somewhat less computer-literate than server users; they can be characterized as "intensive frontier users". So the market is looking at a good performance-to-price ratio.
Low Price Computers	N +. KM	LPC is a mass market; users have no particular skills except in the case of intermediation by a "prescriber".
PCT	N + KM	PCT and players are relatively mass markets, but some advanced users (more in the PCT field and particularly in the PDA market) can play a constructive role in the development of new features.
Players	N	
Platform producers	KM + N	For the OS, as for hardware components, most of the end-users buy a computer with an OS already installed. So the actual users in our sense of the term are computer manufacturers, service companies and sophisticated end-users capable of installing an alternative operating system for their proper use or the use of their customers. On other platforms (database, middleware), the users are also computer manufacturers, service companies and highly-skilled users.
Business solution producers	VH/KM depending on the markets	In the business solutions market, users are professionals. They are able to make a technical evaluation of the product, to carry out trials and tests. This means that people may have skills in the functional domain (what they want, how the software works), and sometimes in the technical one (able to adapt or develop software to meet their own needs, especially in the tools for computer professionals market).
Architects	N (+VH)	Large firms and organizations include very sophisticated users (IT division). SMEs or corporate divisions, at local or sectorial level, are clients of very heterogeneous but rather low IT skills. However, clients may be quite precise in the definition of the services they need, and so in the specification of the application characteristics.

Empirical observation about firms' involvement in FLOSS development can be summarize as so: in the fields where dominant user's skill is either high or very low, firms have invested into FLOSS. When dominant user's skill is intermediate, the dominant design remains that of the classical proprietary model. More precisely, when dominant user's skill is low, competition is price-based and FLOSS helps to provide a cheap solution. When dominant user's skill is high, competition is on quality, services and scalability, and

[30] As explained by Slatter (1992), one of the main strategies for newcomers in technological markets is technological differentiation. Basing its offer on new FLOSS products can be seen as a way for new service companies to differentiate.

[31] In 2005 Gartner forecasted that « 2008, 95 percent of Global 2000 organizations will have formal open-source acquisition and management strategies » (http://www.gartner.com/DisplayDocument?doc_cd=125868)

FLOSS because it is modular, helps to design (so with complementary investments from firms), a better offer. But between this two polar cases, for dominant user rewarding quality for a low-medium price, FLOSS may not be a good alternative to proprietary solutions.

To validate these finding, in the next section, we present a theoretical competition model in which two firms with contrasting attitudes toward intellectual property protection compete on a market characterized by a given distribution of users' skills. We will see how this distribution may drive both the nature (price vs quality) and the issue (market shares and profit levels) of the competition. For this purpose we will give a formal conceptual base to the notion of dominant user's skill, following the underlying assumption that users' skills distribution usually takes a unimodal shape, centred around a given level of skill.

3. A competition model with users' feedbacks

Our ambition, in this paper, is to build a unified framework that could embrace both market and users feedback aspects for understanding why firms present such a wide diversity of their degree of involvement in FLOSS dynamics. Our assumption, fleshed out by the empirical discussion in the section 2 above, in that, beside more usual arguments like the intensity of the competition and the place of software towards the core competences of the firm, users' skills represent a major driver of such a variance. The reason is that users' skills are likely to explain both how users may valorize the intrinsic performances of an information technology system and how they can generate more or less important feedbacks likely to contribute to the improvement and evolution of FLOSS products. This implies deciding consequences, on the one hand, on the price elasticity of the demand and, on the other hand, on the building of the quality of the FLOSS products. Such balance between market and quality effects has to be understood in relation to the distribution of users' skills, or to what we call the dominant user's skill, in the related markets, in a competition context.

This is why we undertook to build a competition model that could catch how the users' skill level may deeply affect the competitive position of a FLOSS involved firm, hence her strategic behaviour when competing against proprietary pattern. The model we develop in this section is a very simple one confronting one proprietary and one open-source software suppliers developing and purchasing partially substitutable products, vertically differentiated by their level of quality. Market is characterized by a given distribution of users' skills that determines both users' reservation price and the capacity of FLOSS customers to generate feedbacks. Budget constraints are not taken into account at this stage.

The users

We consider each individual user to be characterized by a given level of skill denoted $\theta \in [0,1]$. In terms of the problem addressed here, this level can be considered as a good proxy of the individual ability to valorise quality: information technology products are endowed with a wide usage potential that is actually exploited more or less fully, according to the user's capabilities. These capabilities are tightly correlated with the user's level of skill in computing: the more skilled, the more he is able to benefit from the technical quality of the product[32], hence the greater his utility for adopting.

In a pretty standard way we use a linear expression of the individual utility obtained, for a user of level θ, from a product of quality $s>0$ sold at price p such as

$$U = \theta s - p \text{ if he buys the product;}$$
$$= 0 \quad \text{otherwise.}$$

So θs is the reservation price of a user of skill level θ for a product of quality s

Users' skills distribution

[32]See for instance Blili, Raymond and Rivard (1998)

It is considered that the weight of the user population is normalized to 1.

A given market is then characterized by the distribution of skills given by a function

$$f: [0,1] \rightarrow R_+ : \quad \int_0^1 f(\theta) d\theta = 1 \qquad (1)$$

The suppliers

Let *l* denote the open-source firm and *m* the proprietary firm.

Both sell a substitutable product vertically differentiated through a quality level $s > 0$.

The proprietary firm invest a given amount $c_m > 0$ and obtains a level $s_m = m(c_m)$ of the quality level of its product. The production function m is positive increasing and quasi-concave. This permits to take account from the two following aspects. First the more a firm invests in quality, the better this quality. Second, the efficiency of this investment is decreasing as it is harder and harder to improve the quality of a production ; more particularly, in the computer market, this needs to develop dedicated adaptation, thus services, which are less productive (see Cusumano, 2004, chapter 2).

Her product is sold at price p_m

The quality s_l of the product delivered by *the open-source firm* depends on both the level of investment c_l of l to its development, creation of ad-ons and complimentary services ... and the amount of the users' feedback effects from its user base. It is sold at price p_l .

In a first step we will describe the question out of any feedback effect, then in a second step we will introduce those later.

It is considered here that s_l represents the additional quality provided by l to an open-source product freely available on Internet. So the actual quality of the open-source product can be written as the sum $S_l = s_l + s_d$ where s_d is the standard quality of the product downloaded on Internet, hence at price $p_0 = 0$.

So the actual utility for the user to adopt the open-source product from l at price p_l is $U = S_l \theta - p_l$. Meanwhile as far as he will accept to pay only for the additional quality s_l the utility taken into account for the competition model is the sole additional utility provided by the open-source product as purchased by l, that is $U_l = s_l \theta - p_l$

Proposition 1 : Production technologies

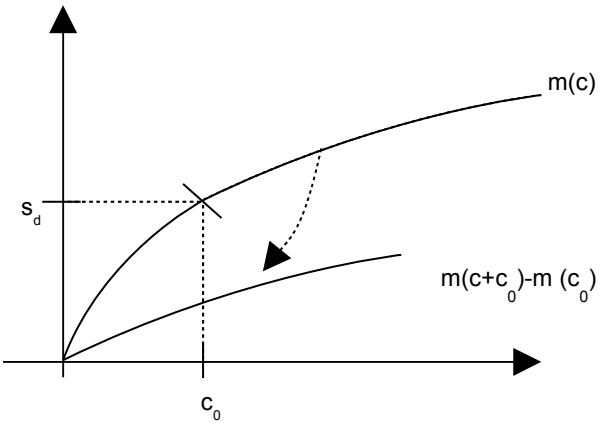
If both firms have the same concave production technology, then the open-source firm gets a lower efficiency when she works on and only on her owns efforts, without drawing benefits from any users' feedback effect.

Proof

Let's consider a production function m twice differentiable increasing and concave, in order to satisfy the characteristics of decreasing returns of the production function.

The proprietary firm reaches the quality level $s_m = m(c_m)$

As far as the open-source firm is concerned, suppose l doesn't make use of any users' feedback effect. Then the conditions of production of an additional quality s_l to the the basic quality s_d of the product freely available on Internet are equivalent to those that would be experienced by m after having reached an equivalent level of basic quality s_d and aiming to produce a better level let's say $s_m = s_d + \Delta s_m$.



m being invertible, $\exists! c_0 \geq 0 : s_d = m(c_0)$ and m being concave it is then possible to write

$$s_l (c)= m (c +c_0) - m (c_0) < m(c)=s_m$$

Thus

$$s_l (c) / c \ < m(c) / c = s_m (c) / c$$

We have then to introduce the users' feedback effect into the production of the open-source quality.

Definition 1: User s' feedback

The users' feedback effect τ_l related to a given market share Θ_l is measured as the cumulated skill level of the user base for the open-source product;

$$\tau_l = \int_{\Theta_l} \theta f(\theta) d\theta \qquad (2)$$

where Θ_l is the user base of l , assumed to be compact.

When taking in account this cumulated skill level of the user base we position our model one step beyond the classical network externalities effects that only consider the size of the related market share.

So investing an amount c_l the open-source firm l reaches, for a market share Θ_l a quality level expressed as

$$s_l = q (c_l , \tau_l) \qquad (3)$$

where τ_l is the amount of the feedback effect as measured by (2).

q is assumed to be twice differentiable increasing and quasi-concave in c_l and τ_l and q(0,.) = 0

Competitive condition

According to proposition 1 and given that the feedback effect is bounded by $\tau_l \leqslant \bar{\tau} = 1$, the condition for allowing the open-source firm to obtain a higher level of quality than the proprietary one can be written

$$\exists\, C > 0 : q(C, 1) > m(C)$$

Denote $C_0 = \text{Inf}\{C > 0 : q(C, 1) > m(C)\}$ (4)

Then

$$\forall c > C_0,\ q(c, 1) > m(c)$$

The linear case

A simple expression of q is given as linear in c_l and τ_l by

$$s_l = \lambda\, c_l\, (1+ \tau_l)\ (5)$$

$$\text{with}\ \ \lambda > 0$$

so that $s_l\,(0, .) = 0$ in so far as when l invests 0, she can only sell what she takes from the community and doesn't provide any added value in terms of interface, adaptation or services that could justify a positive price $p_l > 0$

Accordingly the level of quality reached by the proprietary firm is given by

$$s_m = \mu\, c_m$$

with $\mu > \lambda$ in compliance with proposition 1

In this linear case, the competitive condition can then be written $2\lambda > \mu$ and this is true for any c.

Proposition 2: Market shares

Given a level of prices (p_l , p_m) and qualities (s_l , s_m)

if $s_l < s_m$ then for any user of level θ choosing the open-source software, $\forall\ \theta' < \theta$, θ' chooses open-source software

if $s_l > s_m$ then for any user of level θ choosing the open-source software, $\forall\ \theta' > \theta$, θ' chooses open-source software

Proof

θ adopts the open source software if and only if $U_l\,(\theta) > U_m\,(\theta)$

$$\Leftrightarrow s_l\, \theta - p_l > s_m\, \theta - p_m$$

when $s_l < s_m$ (and $p_l < p_m$)

$$\Leftrightarrow\ \theta \leqslant \frac{p_m - p_l}{s_m - s_l}$$

which remains true for any $\theta' < \theta$

and symmetrically for $s_l > s_m$

Corollary:

Let $\tilde{\theta}$ denote the value of θ at which a user has no preference between l and m

$$\tilde{\theta} = \frac{p_m - p_l}{s_m - s_l} \qquad (6)$$

$\tilde{\theta}$ is called *the indifferent agent*.

Let's also denote θ_0 the minimal level of the users entering the market.

When $s_l < s_m$, $\theta_0 = \frac{p_l}{s_l}$ then $\Theta_l = [\theta_0 , \tilde{\theta}]$ and $\Theta_m = [\tilde{\theta} , 1]$

When $s_m < s_l$, $\theta_0 = \frac{p_m}{s_m}$ then $\Theta_l = [\tilde{\theta} , 1]$ and $\Theta_m = [\theta_0 , \tilde{\theta}]$

Definition 2: The dominant user's skill

As seen before, each market in the software industry is characterized by a dominant type of user that we have described through a typology as "N", "K.M." or "V.H." according to their skill level in relation to the software product. Of course, this dominant type coexists with users of different skill levels, but the distribution can be considered as relatively unimodal. Hence, we characterize the level of the dominant user not through the mean value of θ among the population but through a stronger **criterion of stochastic dominance,** meaning that if f and g are the distribution of users' skills characterizing two markets, then

$$g \gg f \quad \Leftrightarrow \quad \forall \ \hat{\theta} \in [0,1[\ , \ \int_{\hat{\theta}}^{1} \theta\, g(\theta)\, d\theta > \int_{\hat{\theta}}^{1} \theta\, f(\theta)\, d\theta \qquad (7)$$

This criterion expresses a shifting of the mass of the users from a lower to a higher level of skills.

The game:

In this model, strategic decisions are taken simultaneously, each agent formulating rational expectations about the behaviour of his competitor. Thus we consider that the level of investment of each firm is decided while taking count of the expected skill level of the indifferent agent thus of the respective market shares and the expected feedback effect for the open-source firm. However we model the strategic interaction of the two firms as a two-stages game, since there are two strategic variables for each $i \in \{l , m\}$, the investment c_i and the price p_i . In the first stage, each firm chooses a level of investment c_i that permits them to get an intrinsic quality level q (c_l , 0) and m (c_m) respectively. The second stage is a price competition in which each firm chooses a price p_i , this price has a recursive effect on the level of the indifferent agent, thus on the respective market shares and the level of the users' feedback for the open-source firm τ_l thus the level of quality she's actually reaching $s_l = q(c_l , \tau_l)$. So that $\tilde{\theta}$ is given through an implicit equation of the strategic variables (c_l , c_m) and (p_l , p_m).

The competitors pay-offs can then be written following the relative level of realized qualities as follows

	Π_l	Π_m
$s_l > s_m$	$[1 - F(\tilde{\theta})]\, p_l - c_l$	$[F(\tilde{\theta}) - F(\theta_0)]\, p_m - c_m$
$s_l < s_m$	$[F(\tilde{\theta}) - F(\theta_0)]\, p_l - c_l$	$[1 - F(\tilde{\theta})]\, p_m - c_m$

Where F is the cumulated distribution of users' skills and $\theta_0 = Min[\frac{p_l}{s_l}, \frac{p_m}{s_m}]$ (8)

Proposition 3 : Low users' skills

When the dominant user's skill is low enough, the open-source firm invests little, and stays at a lower level of quality than the proprietary one , targeting a relatively price-sensitive market.

Proof:

$\forall\ \tilde{\theta}$ defined as above, $\tau_l = \int_{\Theta_l} \theta f(\theta) d\theta \leqslant \int_0^1 \theta f(\theta) d\theta = \tau$

Thus

$$s_l = q(c_l, \tau_l) \leq q(c_l, \tau)$$

According to Proposition 2

$$q(c_l, 0) < m(c_l)$$

q being continuously increasing, there exists $\tau_0 > 0$ such as $\forall \eta < \tau_0$, $q(c_l, \eta) < m(c_l)$

(In the linear case, such condition can be written $\tau < \frac{\mu}{\lambda} - 1$)

Then, the proprietary firm may fix a price p_m generating a positive profit for her while preventing the open source firm from targeting a higher or equal level of quality with non negative profits.

So l targets a low level of quality, fixes a low price and maximise her profit by enlarging as much as possible her market share $[\theta_0, \tilde{\theta}]$.

Proposition 4:

If the feedback effect is efficient enough, when the dominant user's skill is high enough the open-source firm may invest much, obtain a higher level of quality as the proprietary one, targeting a relatively quality sensitive market. The proprietary firm invests less, aims a lower level of quality and targets a relatively less skilled and more price sensitive market segment.

Proof :

If all the population is concentrated at skill level 1, then the global feedback is maximal at the value $\bar{\tau} = 1$

Let's suppose l wins the whole market.

At level of investment c_l , l reaches the quality $s_l = q(c_l, 1)$

$\forall c_l \geq C_0$, then $s_l > m(c_l)$

The pay-off of l can be written $\Pi_l = p_l - c_l > 0$ if and only if $p_l > c_l$

For any user, the utility of adopting the FLOSS product is $U_l = s_l - p_l = q(c_l, 1) - p_l$

then $U_l > 0 \Leftrightarrow p_l < q(c_l, 1)$ ($p_l < 2 \lambda c_l$ in the linear case)

So l may fix her price p_l under the double constraint $c_l < p_l < q(c_l, 1)$ (9)

what requires $q(c_l, 1) > c_l$ ($\lambda > ½$ in the linear case).

Entry threat :

m may threaten the l's monopoly and capture the whole market if and only if there exists a level c_m of m's investment and a price p_m such that

$$U_m(c_m, p_m) > U_l(c_l, p_l)$$

$$\Leftrightarrow \quad m(c_m) - p_m > q(c_l, 1) - p_l$$

and

$$\Pi_m(c_m, p_m) > 0 \Leftrightarrow p_m > c_m$$

as for l, m has to fix her price under the double constraint $c_m < p_m < m(c_m)$ (11)

what requires $m(c_m) > c_m$ ($\mu > 1$ in the linear case).

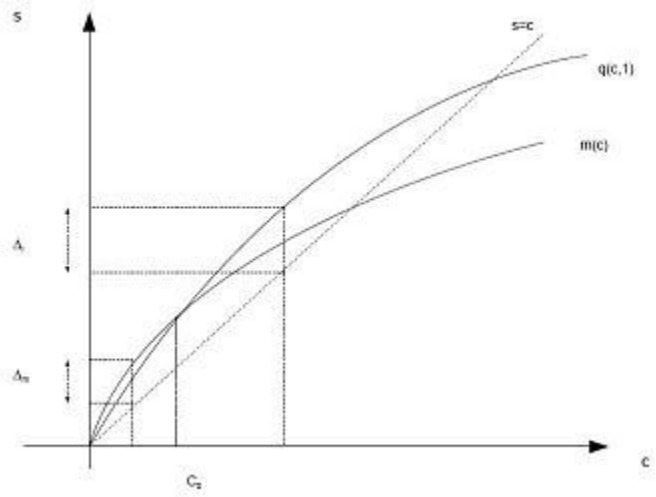
As far as $q(c_l, 1) - c_l > m(c_m) - c_m$,

l may prevent this threat of entry by fixing a price $p_l < q(c_l, 1) - (m(c_m) - c_m)$

In the linear case, $m(c) - c$ is growing with c so it would be possible for m to capture the market by investing at a high enough level c_m . But when introducing budget constraints from the users side this doesn't hold any more as far as l invest at a high enough level.

In the concave case conversely, the differential $m(c) - c$ is decreasing with high c . m may capture the market, by targeting a lower quality price if and only if

$$\Delta_m = \max (m(c) - c) > \Delta_l = \max_{l \geq c 0} (q(c_l, 1) - c_l)$$



Δ_m , respectively Δ_l correspond to values of c giving a slope equal to one for m(c) and q(c,1) respectively and their respective widths depend on the respective concavity of the two curves.

So the more efficient the feedback effect, the more the open-source curve moves away from the diagonal and the open-source firm may deter the proprietary threat to entry.

Even if the proprietary firm cannot enter the market, this potential threat prevents l to fix a monopoly price.

More generally, when the dominant user's skill is high, if the production technology conditions and budget constraints allow it, the open-source firm invests much, targets a high level of quality and a quality sensible market.

In conclusion, we can say that if the dominant user is unskilled, and so price-taker, FLOSS may be a good strategy if it enables the firm to supply a product at a lower price than the proprietary one, but requiring a low level of investment to adapt or improve the FLOSS solution. In the opposite situation, when the dominant user is very skilled, an offer based on FLOSS and a high level of complementary services can succeed. This result is in accordance with the work of Henkel (2006), who shows in a duopoly model that "a regime with compulsory revealing can lead not only to higher profits, but also to higher product qualities than a proprietary regime". In the intermediate situations, the FLOSS strategy may be less efficient than the proprietary one.

4. Industrial and managerial issues.

The model we have introduced in section 3 permits us to enlighten how the distribution of users' skills on the different markets can contribute to explain the variance of firms' involvement into FLOSS. At the heart of this question lies the possibility to arouse and mobilize feedbacks from the users and to appropriate them as an input of the firm production function.

The basic assumption here is that the more skilled the users are, the more they are likely to provide different forms of returns that can be used for improving the product, so the more they can be considered as a value-added source that may play a role in the firm's business model.

This question has undoubtedly something to deal with that of the relations between firms and communities. But it is much wider in so far as it embraces all the types of users that can be found on the different markets. Beyond communities considered as formed by developers contributing to the development-improvement of a product they usually make and use, the feedback effects an open-source firm has to absorb are those from their customers. These latter do not necessary behave as community members and are not always capable of contributing to software development. So, in the open source model, firms have to play a coordination and go-between role with the contributing users, coordinating, making coherent their contributions, but also to manage assistance and specific services to the use for the less skilled or the ready to pay customers.

There is a reciprocal effect between the open-source firm and her customer base. In the one side, the open-source firm has to internalise the feedback effects from the users into her production process in order to improve the quality of its commercial offer. In the other side, customers consider that, as far as the firm benefits from positive feedbacks from her customer base and from open-source community, she may provide a product endowed with good properties as summarized by what we called the "3A services" (assistance, assurance and adaptation to the use).

In application markets, all the recent new entries have been based on the competitive advantage drawn from the FLOSS label: FLOSS OS editors (like RedHat), FLOSS database producers (MySQL), FLOSS service companies (VA Linux, or Linagora in France). Today, incumbents are also assuming this strategy (IBM with Eclipse, SAP opening its data base system, even Microsoft opening some of its technical tools...) This could shortly become the benchmark of industrial organization on these markets, inducing a growing control of FLOSS development by commercial firms and a spectacular enlargement of open IP regimes in the software field.

But there is a border where the consumers' feedback become too weak to play a significant role for improving enough the quality. When it is the case, they do notcontribute enough to the open-source firm's competitive advantage, throwing her back, as formalized in the model, to a less efficient situation as her proprietary competitors.

What the model also shows in that case, (where the users are not skilled enough to permit the open-source firm to provide a better quality product than the proprietary one), is that the FLOSS firm will shift to an alternative market strategy. She will target the less skilled customers and enlarge the market base towards users with lower reservation prices. This is also a traditional behavior in the computer industry, where successively Digital entered the market creating the mini-computers thanks to the transistors, and Apple did, creating the personal computer thanks to the micro-processor. Linux and Open source technologies would play the same role making possible the provision of under $200 notebooks and under $100 computer for Third World countries MIT project. The limit situation consists in purchasing the standard product as freely downloadable on Internet but ready to run for the user, avoiding him heavy searching costs, all the more worrying if he is unskilled. So the open-source firm value-added lies for those customers in providing them with the appropriate product and complementary services like training, assistance and advices. This is, for instance, the business model RedHat or Mandriva developed at the beginning, with the Linux distribution on CD, and a model already announced by Ousterhout (1999). It also appears close to the common explanation that the FLOSS product steps in as a complement to what the firm actually sells, either hardware (Asus with Eee PC) or services (Aol-Netscape sponsoring Firefox Browser to give access to its information and entertainment services) and it is to be said that we do not know successful stories based on a pure strategy of that kind.

Of course, one might oppose the argument that such strategies have a flavour of free-riding and can yield to demotivate the most involved people in the communities, seeing others drawing profits from their own benevolent work[33]. However, by adopting FLOSS products, those firms take part to the enlargement of FLOSS users' network related to a given open software like the operating system Linux or the office application suite “Open Office”. In a competition regime in which the battle for network externalities and then standards play a crucial role, this can be considered as a strong help in favour of FLOSS, that can win at least community's neutrality if not kind approbation.

Our model bridges those two polar situations where the strategy of the open-source firms is respectively targeting either quality or price sensible markets. It also echoes an old managerial choice firms have to do entering the computer market (Cusumano 2004): either being a service company or a product company. Observable FLOSS strategies and our model converge confirming what Cusumano explained: 1) if there is a market for service, ie if customers are ready to pay for it (because they understand the added value of this service), it is worth doing it. But if not, firms must concentrate on selling quite standard products and to bet on the economy of scale. 2) there is a point where it is better stopping investing in services, because the costs are growing quicker than the market share or the willingness to pay.

In between, the opportunity for an open-source strategy will not vanish for all that, but will tightly depend on the technical nature of the product and the production process, more particularly the possibility to take advantage of medium skilled users base. One crucial aspect in this concern will be the importance of test tasks facing the development of the product. Kogut and Metiu (2001) show that these “frontier-users” not skilled enough to contribute by code writing are nevertheless capable to bring a decisive contribution to software improvement by constituting a huge test and debugging base in a field where maintenance costs can reach 50 to 80% of the software budget…An interesting illustration is given by automate crash description programs included in most of the popular desktop FLOSS products (Firefox, Open Office).

When software is not a core competence of the firm, like it is mostly the case for personal communication tools, or low-medium quality computers, the firm who has rallied to FLOSS has a huge interest to bridge her base of low and medium skilled users with the communities involved in related software tools development. In one way, she may help the users to find their path in an always widening library of open-source applications. In the other way she might pay attention to the expression of requirement, satisfactions and dissatisfactions of her customers in order both to early identify new needs and the new features they could justify and to reverberate the appropriate informations more or less actively within the concerned communities of developers.

This is a precious lesson for understanding open innovation model. Even the dominant user's skill being high or medium level, these potential feedbacks still have to be organised, coordinated and integrated into the

[33]See Foray, Thoron and Zimmermann (2007) for a discussion of that particular point.

open-source firm production process. First open-source firms have to organise customer services capable to consider users not only as a demand for maintenance or assistance services, but also as a valuable source of information capable to play a crucial role in a partly decentralized open model of innovation.

Second, for being able to integrate knowledge and innovation from the open-source communities, open-source firms have to develop internally efficient capabilities of absorption, essential condition to capitalize and internalise the communities' contribution and the users' feedbacks for improving their own product quality. Dahlander and Magnusson (2008) working on the relations between firms and open-source communities show that those firms need "to develop sufficient absorptive capacity to benefit from external developments, not only to identify useful external knowledge, but also to assimilate and apply it". This is reflected in the model by the partial complementarity of the feedbacks with the firm efforts within the production function q, requiring positive investment to give a positive outcome whatever the level of users' feedback - q(0, .) = 0-.

This corresponds to the more general assertion from Cohen and Levinthal (1990) about the necessity for a firm to make internal efforts of R&D as a prerequisite for the absorption of external technology. We may formulate the hypothesis that the more skilled the users are, the more involved the firm must be, because of the growing level of complexity of the feedbacks and of the demand. The ultimate case, illustrated by AdaCore Technology or MySQL being that managing of the open source software community is the specific asset of the firm.

Nevertheless, the origin of the open-source rationale remains that of developer-users pooling their development efforts for their own needs, aiming at better access to efficient tools for everyone. This is the core of the open-source model and, as seen in the paper, the recent move towards commercial structuring doesn't contradict this foundation : either this volunteer collaboration is initiated by individuals or by a firm, a key condition of the success is users-developers adhesion. The management of the relation between an open-source firm and the related communities is of crucial importance as a recent literature has already emphasized. In that respect, another interesting contribution that enlightens interestingly our concern has been done by Âgerfalk and Fitzgerald (2008) that did study how firms dealing with communities engage themselves in open source governance. The dominant characteristic the authors have observed aims to preserve the co-existence and co-operation of two types of organisation that are based on remote albeit not contradictory rationales, in a nutshell "not seeking to dominate and control process", "providing professional management and business expertise" and "helping establish an open and trusted ecosystem". They view such interaction as rather osmotic than parasitic, as firm's resources reinforce communities sustainability.

These considerations could be easily extended from the sole framework of the communities to the whole users base. In this paper, we have insisted on the role that the dominant user's skill plays to permit firms to invest into FLOSS development helping them to offer better quality products. In fact, this causality could be interestingly reversed taking into account that FLOSS, by mobilizing users expression and creativity in an open context, helps these latter to better develop their skills. Proprietary software does not[34].

The FLOSS movement has sometimes been presented as a canonical model of production for the open innovation paradigm, and even for the knowledge society. If so, open development may develop in fields where users are skilled enough to initiate the development of open knowledge and have enough market power to force the traditional producers to shift to an open model. In such conditions, the open IP regime can be seen as a very efficient solution to the Schumpeterian dilemma in so far as it permits a wide diffusion of knowledge, while encouraging innovation, as producers are incited to contribute to the development of the product they use/sell. This regime could be named VH open innovation regime, in reference to Von Hippel (1988)'s seminal work on users as innovators. Open initiatives have been launched in many industries, such as biotech, remote sensing and chip design. Most of the time, their chances of success are evaluated in terms of the motivation of the participants and the stability of the "community". Our contribution argues for more economic aspects of the evaluation, by taking into account the impact of the users in these productions, and their bargaining power.

[34]We thank sincerely the anonymous referee who made us this suggestion.

References.

P.J. Ãgerfalk, Fitzgerald B.T. 2008, "Outsourcing to an unknown workforce: Exploring opensourcing as a global sourcing strategy", *MIS Quaterly*, vol.32, n°2, pp. 385-409

J. Bessen and R.M. Hunt, 2007, "An Empirical Look at Software Patents", *Journal of Economics and Management Strategy*, Volume 16, Number 1, Spring, 157-189.

S. Bhattacharya and Guriev S, 2006. 'Patents vs. trade secrets: Knowledge licensing and spillover.' *Journal of the Economic Association* 4, no. 6, pp. 1112-1147.

S. Blili, L. Raymond, S. Rivard, 1998, "Impact of task uncertainty, end-user involvement, and competence on the success of end user computing", *Information Management* **33** (3) 137-153.

M. Boldrine and D.K. Levine, 2007, "Open Source Software, Who Need Intellectual Property?", *The Freeman, Ideas on Liberty*, Vol. 57, N°1, January, Foundation for Economic Education.

A. Bonaccorsi, Giannangeli S. and Rossi C. 2006. Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry, *Management Science* Volume 52 , Issue 7 (July 2006), pp 1085-1098

M. Campbell-Kelly and Garcia-Swartz, D. G From Products to Services: The Software Industry in the Internet Era. Available at SSRN: http://ssrn.com/abstract=984451

V. Chang, Mills, H. and Newhouse, S., 2007 From Open Source to long-term sustainability: Review of Business Models and Case studies. In: *All Hands Meeting 2007, OMII-UK Workshop*, 10 September - 13 September, 2007, Nottigham, UK.

H. Chesbrough, 2003. *Open innovation: The New Imperative for Creating and Profiting from Technology.* Harvard Business School Press, Boston MA.

H. Chesbrough, 2006. Open innovation: a new paradigm for understanding industrial innovation. In: Chesbrough, H., Vanhaverbeke, W., West, J. (Eds.), *Open Innovation: Researching a New Paradigm*. Oxford University Press, Oxford.

M.W.Cohen and D.A.Levinthal, 1990, "Absorptive Capacity: a New Perspective on Learning and Innovation", *Administrative Science Quarterly*, (35) 128-152.

J. Crémer, d'Aspremont C. and Gérard-Varet L.-A., 1990. « Incentives and the Existence of Pareto-Optimal Revelation Mechanisms », *Journal of Economic Theory*, vol. 51, n. 2, august, pp. 233-254.

M.Cusumano, 2004, *The Business of software: what every manager, programmer, and entrepreneur must know to thrive and survive in good times and bad* , New York, Free Press

L. Dahlander and M. W. Wallin, 2006**.** A man on the inside: Unlocking communities as complementary assets. *Research Policy*, 35, 1243-1259.

L. Dahlander and M. G. Magnusson, 2005**.** Relationships Between Open Source Software Companies and Communities: Observations from Nordic Firms. *Research Policy*, 34, 481-493

L. Dahlander and M. G. Magnusson, 2008. How do Firms Make Use of Open Source Communities? , *Long Range Planning* 41 (2008) 629e 649

J. De Bandt, 1995. *Services aux entreprises: informations, produits, richesses*. Economica, Paris.

D. Demazière, F. Horn and N. Jullien, How Free Software Developers Work: The Mobilization of 'Distant Communities'(July 2006). Available at SSRN: http://ssrn.com/abstract=1301572 A former version (in French) has been published in *Cahier lillois d'économie et de sociologie*, 2005, n°46, pp 171-194.

J. Farrell and G. Saloner, 1985. Standardization, compatibility and innovation. *Rand Journal of Economics* 16 , pp. 70–83.

D. Foray, S. Thoron and J.-B. Zimmerman, 2007 "Open Software: Knowledge Openness and Cooperation in Cyberspace", in Brousseau E. and Curien N. (Eds.), *Internet and Digital Economics; Principles, Methods and Applications*, Cambridge University Press, 2007.

L-A. Gérard-Varet and J-B. Zimmermann, 1985, Concept de produit informatique et comportement des agents de l'industrie, Colloque "Structures Economiques et Econométrie" - Lyon, 23 - 24 mai 1985.

J.Henkel, 2006, « The Jukebox Mode of Innovation – A Model of Commercial Open Source Development », DRUID Working Paper 06-25.

F. Horn, 2004. *L'économie des logiciels*, collection repères, la Découverte, 128 p.

M. Iansiti and G. L. Richards, 2006. The Business of Free Software: Enterprise Incentives, Investment, and Motivation in the Open Source Community. Harvard Business School Working Paper Series, No. 07-028.

B. M. Kogut and A. Metiu, 2001. Open Source Software Development and Distributed Innovation, *Oxford Review of Economic Policy*, Vol. 17, No. 2, Summer 2001.

H. Koski, 2007, OSS Production and Licensing Strategies of Software Firms. *Review of Economic Research on Copyright Issues,* 2005, vol. 2(2), pp. 111-125

K. Lakhani and E. von Hippel, 2003. How open source software works: Free user to user assistance. *Research Policy*, 32: 923-943. URL: http://opensource.mit.edu/papers/lakhanivonhippelusersupport.pdf.

K. Lakhani et R. Wolf, 2005. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects, in Feller, J., R. Fitzgerald, S. Hissam, & R. K. Lakhani (Eds.). *Perspectives on free and open source software* , MIT Press.

L. Muselli, 2002, Licenses: strategic tools for software publishers? In Clément-Fontaine et al., [2002**]**, editor, *New Economic Models, New Software Industry Economy*, pages 129-145.

J. Ousterhout. 1999. Free Software Needs Profit. C*ommunications of the ACM,* Volume 42 , Issue 4 (April 1999), pages: 44 - 45

S. Slatter, 1992. *Gambling on Growth: How to Manage the Small High-Tech Firm*, Wiley, New York.

W.E. Steinmueller, 1996, 'The U.S. Software Industry: An Analysis and Interpretive History', in D. Mowery (ed.), *The international computer software industry: a comparative study of industry evolution and structure*, Oxford University Press:Oxford and NY.

D. J. Teece, 1986, 'Profiting from technological innovation: Implications for integration, collaboration, licensing and public policy,' *Research Policy*, 15(6), 285–305.

E. Von Hippel, 1988, *The Sources of Innovation*, Oxford University Press.

E. Von Hippel, 2002. Open Source Software as horizontal innovation networks – by and for users, MIT Sloan School of Management W.P. N°4366-02.

E. Von Hippel, 2005. *Democratizing Innovation*, Cambridge, MA: MIT Press (April).

E. Von Hippel and G. von Krogh , 2003. Open Source Software and the "Private-Collective" Innovation Model: Issues for Organization Science . *Organization Science*/Vol. 14, No. 2, March–Apri, pp 209-223.

J. West, 2003. How open is open enough? Melding proprietary and open source platform strategies. *Research Policy* 32 (7), 1259-1285.

J.-B. Zimmermann, 1995, Le concept de grappes technologiques. Un cadre formel. *Revue économique*, 46 (5): 1263-1295.